

## DATA MANAGEMENT METHOD FOR DATABASE SYSTEM

Publication number: JP2002157158

**Publication date:** 2002-05-31

**Inventor:** KAWAKAMI NOBUHIKO; KIKUCHI SATOSHI; SHIGA KENTA; YUI HITOSHI

**Applicant:** HITACHI LTD

### Classification:

- International: **G06F17/30; G06F12/00; G06F17/30; G06F12/00;**  
(IPC1-7): G06F12/00; G06F17/30

### - European:

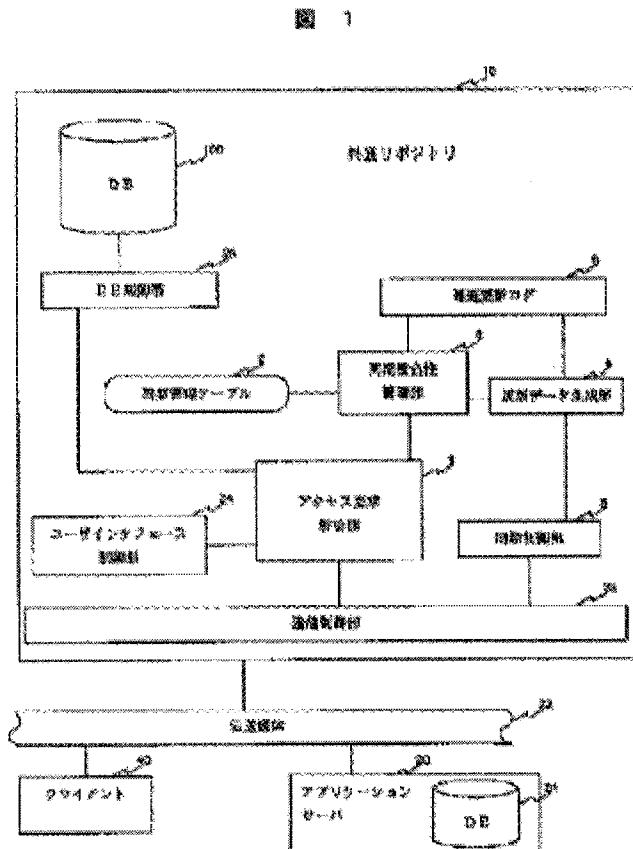
Application number: JP20000357622 20001120

**Priority number(s):** JP20000357622 20001120

[Report a data error here](#)

## Abstract of JP2002157158

**PROBLEM TO BE SOLVED:** To improve convenience of a system in data management between plural databases by preventing generation of inconsistency caused by a difference in synchronous timing when synchronizing update processing. **SOLUTION:** In a synchronizing method between the databases, data update is managed in a main database, and updated data are synchronized to one or more sub databases. In the synchronizing method, the immediate or delayed synchronous timing is set, and the synchronous processing is executed according to the set conditions. By associating contents of required immediate processing with the updated data recorded in a log as delayed processing, synchronous data wherein the necessary updated data are lumped is generated, and the synchronous data is synchronized to the sub databases.



Data supplied from the [esp@cenet](mailto:esp@cenet) database - Worldwide

**Family list**

1 family member for: **JP2002157158**

Derived from 1 application

[Back to JP2002157](#)

**1 DATA MANAGEMENT METHOD FOR DATABASE SYSTEM**

**Inventor:** KAWAKAMI NOBUHIKO; KIKUCHI  
SATOSHI; (+2)

**Applicant:** HITACHI LTD

**EC:**

**IPC:** *G06F17/30; G06F12/00; G06F17/30* (+3)

**Publication info:** **JP2002157158 A** - 2002-05-31

---

Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許序 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号  
特開2002-157158  
(P2002-157158A)

(43)公開日 平成14年5月31日(2002.5.31)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テ-マ-ト <sup>8</sup> (参考)
G 0 6 F 12/00	5 3 3	G 0 6 F 12/00	5 3 3 J 5 B 0 7 5
	5 4 5		5 4 5 A 5 B 0 8 2
17/30	1 1 0	17/30	1 1 0 C
	2 4 0		2 4 0 A

審査請求 未請求 請求項の数3 O.L (全 14 頁)

(21)出願番号	特願2000-357622(P2000-357622)	(71)出願人	000005108 株式会社日立製作所 東京都千代田区神田錦河台四丁目 6 番地
(22)出願日	平成12年11月20日(2000.11.20)	(72)発明者	川上 順彦 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
		(72)発明者	菊地 聰 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
		(74)代理人	100075096 弁理士 作田 康夫

最終頁に統ぐ

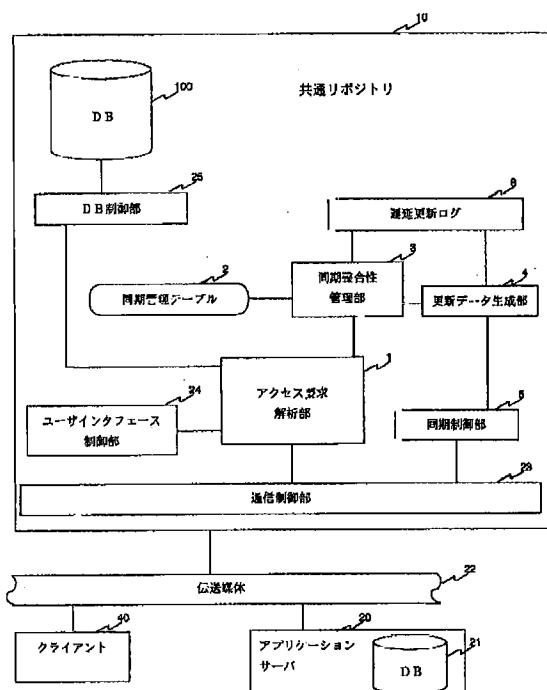
(54) 【発明の名称】 データベースシステムにおけるデータ管理方法

（57）【要約】

【課題】複数データベース間のデータ管理において、更新処理の同期を行うにあたり同期タイミングの相違による不整合性発生を防止することでシステムの利便性を向上する。

【解決手段】主データベースにおいてデータの更新を管理し、1つ以上の従データベースへ更新データを同期するデータベース間の同期方法として、即時、或いは遅延の何れかの同期タイミングを設定しておき、その設定条件に従って同期処理を行うにあたって、要求される即時処理の内容と遅延処理としてログに記録された更新データを関連付けることによって、必要な更新データを一括した同期データを生成し、前記同期データを従データベースへ同期する。

1



【特許請求の範囲】

【請求項1】 複数のサーバが伝送媒体を介して相互に接続され、第1のサーバを主データベースとしてデータの更新を管理すると共に、第2のサーバを従データベースとして、更新したデータを少なくとも1つの従データベースに管理されるデータに同期する機能を備え、前記主データベースに格納されるデータの更新を要求する処理種別に対応する同期方法として、更新処理発生時に即時に同期処理を行うか、或いは遅延して同期処理を行うかをあらかじめ設定しておくことで、その設定条件に従って主データベースから従データベースへのデータの同期処理を行うデータベースシステムにおいて、前記同期処理を実現するにあたって、設定された同期方法と要求される更新処理の内容を基に、即時同期設定と遅延同期設定の2つの異なる同期方法をもって処理される複数の更新処理を関連付けることによって、遅延同期処理の対象として主データベースに蓄積された更新データに関しても、即時同期処理の対象として主データベースに要求された更新処理の同期に伴って一括して同期処理を実行することを可能とし、データベース間の整合性を保証した同期処理を実現するデータ管理手段を有することを特徴とするデータ管理方法。

【請求項2】 前記前記主データベースで行われたデータ更新を従データベースに同期する方式を設定するため、主データベースに対して要求された更新種別を判定するアクセス要求解析手段と、前記主データベースへの更新処理の内容に対応する同期方法の設定を記憶する同期管理テーブル手段と、前記テーブルにおいて遅延同期方法を適用された更新内容を従データベースに同期するまで主データベース上に蓄積しておく遅延更新ログ手段、従データベースへの同期データを生成する更新データ生成手段と、前記更新データ生成手段によって生成されたデータを基に、設定された同期スケジュールに基づいて従データベースにデータを同期する同期制御手段を有するデータベースシステムにおいて、前記ログ手段及び前記テーブル手段の設定を基にして、主データベースに対して発生した更新処理種別と処理対象となる属性、同期対象オブジェクト、及び同期タイミングの関連付けを行う同期整合性管理手段を有することを特徴とするデータ管理方法。

【請求項3】 請求項1及び請求項2に記載のデータ管理方法において、

データベースに設定された同期方法と要求される更新処理の内容を基に、即時同期設定と遅延同期設定の2つの異なる同期方法をもって処理される複数の更新処理を関連付けることによって、データベース間の整合性を保証した同期処理を実現するデータ管理手段を、クライアントに備えることを特徴とするデータ管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データベースシステムにおけるデータベース間のデータ同期を実現するデータ管理方法に関するもので、特に、第1のサーバにおいてデータを一元的に管理し、更新データを同期設定に基づいて第2のサーバが管理するデータに同期することを行うデータ管理方法及びデータ管理システムに関するもの。

【0002】

【従来の技術】企業情報システムなどの大規模情報システムにおいては、社員の個人データ、組織データなどを管理するため、あるいは業務関連の情報の蓄積や共有のために広くデータベースが用いられている。このような目的で使用されるデータベースの1つとして、ディレクトリシステムが挙げられる。

【0003】例えばディレクトリシステムは、電子メールの送信相手の姓名などからメールアドレスを検索する、所謂電子電話帳機能として利用されている。

【0004】ディレクトリシステムは、人、組織、ネットワーク機器など管理対象となる情報を木構造で表現する。この木構造データモデルは、ディレクトリシステムに関する標準として、CCITTの勧告であるX.500 (ISO9594) によって規定されたものである。木構造の枝葉に相当する箇所には、ディレクトリエンティが配置され、これが人や組織を表すデータ単位となる。各々のエンティは、ディレクトリツリーのルートからのパスを示す名称であるDN (Distinguished Name) によって一意に識別される。さらに、各々のエンティはユーザのメールアドレスに加え、姓名、電話番号、FAX番号など、様々な情報を属性として記憶可能である。

【0005】また、X.500はディレクトリシステムをクライアント・サーバシステムとして規定しており、ディレクトリクライアントは、前記ディレクトリデータに対してDNによる読み出し及び更新、ディレクトリエンティのデータである属性をキーとした検索が可能である。このデータの読み出しや更新、検索の時に使用されるクライアント・サーバ間の通信プロトコルとして、OSI (Open Systems Interconnection) のアーキテクチャに基づくDAP (Directory Access Protocol) が規定されている。しかし、このDAPはOSI層を必要とする負荷の高いプロトコルであることから、その使用が倦怠されてきた。

【0006】そこで、アメリカのミシガン大学が、より軽量なディレクトリアクセス用のプロトコルとしてLDAP (Lightweight Directory Access Protocol) (RFC2251) を開発した。LDAPはDAPのサブセットであり、T

CP/IP スタック上で動作し、インターネット標準化団体である IETF (Internet Engineering Task Force) により標準化された。

【0007】ところで、昨今の企業情報システムにおいては、様々な業務を行うために必要に応じて電子メールシステムをはじめ、ネットワークOSやグループウェアなど、種々のアプリケーションが導入されている。しかし、これらのアプリケーションは、ユーザ情報、組織情報などのデータを個別に管理する必要があった。そのため、システム管理者は人事異動や組織改編などのデータ更新の度に、情報システム内で更新を必要とする全てのアプリケーションのデータを管理しなければならず、その運用負荷は増大する一方であった。

【0008】例えば、人事データベースとグループウェアの双方で社員氏名、社員コードを管理するなど、情報システム内の各アプリケーションが管理するデータには重複した属性が多く存在するが、これらの属性を管理するアプリケーションの多くは、ユーザ情報を独自のデータベースを用いて独自の形式で管理しているのが現状であった。

【0009】そのため、運用対象となる複数のアプリケーションデータを一元管理するメタディレクトリシステムが注目されている。メタディレクトリシステムは、ディレクトリシステムに対して行われた更新を自動的にアプリケーションのデータベースへ反映させることにより、ディレクトリシステムに登録されているユーザ情報、組織情報をはじめとする各種資源情報と、各アプリケーションのデータベースに登録されている情報との同期をとる。これにより、ユーザがパスワードの変更を行う時に全てのアプリケーションに対して変更処理を行うといった複数のアプリケーションに重複した属性の個別管理の必要がなくなり、ユーザのデータ管理負荷を軽減できる。

【0010】前記メタディレクトリシステムは、同期対象となる複数のアプリケーションやアプリケーションが管理する属性に対して個別に同期タイミングを設定可能である。

【0011】このようなデータの一貫性管理方法の従来技術として特開平11-219309号公報が挙げられる。

【0012】前記一貫性管理方法においては、主データベースと複数の従データベースの間で、全ての処理に対するデータの同期を即時に行うことで厳格な一貫性管理を行うか、或いは遅延して同期することにより厳格性を緩和した一貫性管理を行うかを選択可能である。

【0013】例えば、アプリケーションでユーザ認証に用いられるパスワードなどの属性は本人認証の度に使用する緊急性の高い属性であるため、この属性が更新された場合にその更新内容は即時に同期する一方で、新入社

員の入社や人事異動などに伴い大量のユーザ情報の登録、更新を行う必要がある場合には、同期の即時性よりも1日の更新データを夜間バッチ処理により一括して同期することでネットワーク負荷を軽減することが可能である。

【0014】

【発明が解決しようとする課題】しかしながら、上記のようなデータベースシステムの一貫性管理方法においては、更新処理の種類或いは更新対象の属性に対応したデータベース間のデータ同期のタイミングを設定することは可能であったが、以下に述べるように、データベース間のデータの整合性が保証されないという問題があった。

【0015】例えば、追加・変更・削除といった更新種別や、その更新対象の属性によって情報システムにおける緊急性が異なるため、それぞれ適切な同期タイミングの設定を行った場合に、同期タイミングの相違により主データベースと従データベース上のデータ間が不整合となることがあった。その結果、システム管理者からの整合性回復作業が必要となり、また、ユーザはデータベース間で異なるデータに基づいて誤った業務を遂行してしまう可能性があり、利便性の面で問題があった。

【0016】具体的に、ユーザや組織などの追加処理についてではネットワーク負荷を低く押さえるために日次で一括して従データベースへ同期し、パスワードなどの緊急性の高いデータの更新処理については即時に同期したいというケースについて説明する。

【0017】この場合、主データベース上にユーザが初期登録された時にはデフォルトのパスワードが付与され、同ユーザが直ちにパスワードの変更を要求すると、同ユーザの更新されたパスワードを従データベースへ即時に同期することとなる。しかし、この時点では主データベース上に追加されたユーザ情報が従データベースに同期されていないため、従データベース上には当該ユーザ情報が格納されておらず、パスワードの同期処理を実行することができない。

【0018】本発明の目的は、ディレクトリシステムをはじめとしたデータベースシステムにおいて、複数のデータベースが稼働している環境における整合性を保証したデータ同期方法を提供してデータベースシステムの利便性の向上を図ることである。

【0019】

【課題を解決するための手段】上記の目的を達成するため、本発明は、複数のサーバが伝送媒体を介して相互に接続された環境でデータベース間の更新内容に対応する同期タイミングをあらかじめ設定可能なデータベースシステムにおいて、データベースへの更新処理に対するデータベース間の整合性管理方針を主データベースに決定しておき、更新処理の発生時に設定に基づいて更新処理種別と処理の対象となるオブジェクト、及び属性を関連

付け、主データベースから従データベースへ更新データの同期を行うデータ管理方法を提供する。

【0020】かかるデータ管理方法を用いることにより、データベースシステムにおける更新データの同期方法として、即時、或いは遅延の何れかの同期タイミングを設定しておき、設定された同期方法を基に、主データベースに要求される即時処理の内容と既に要求された処理が遅延処理としてログに記録された更新データを関連付けることによって、必要な更新データを一括した同期データを生成すると共に従データベースへ同期して、データベース間の整合性を保証した同期処理を実現する。

【0021】また、本発明は、上記主データベースで行われたデータ更新を従データベースに同期する方式を設定するために、主データベースに、同期対象データベース毎に個別の設定を行うための同期データベース管理を行うと同時に、更新処理の種別、処理の対象となる属性毎に適用する同期タイミングを記憶するテーブル手段と、データの遅延同期を行うために遅延同期の対象として設定された更新処理の内容を記録するログ手段と、主データベースに対して行われた更新データを従データベースに同期するための更新データの生成手段と、前記更新データの従データベースに対する同期処理を実行する手段を有するデータベースシステムにおいて、主データベースに対して発生した更新処理種別と処理対象となる属性、同期対象オブジェクト、及び同期タイミングを関連付ける手段を有するデータ管理方法を提供する。

【0022】かかるデータ管理システムを用いることにより、前記データベース間の整合性を保証した同期処理を実現するデータ管理方法を提供することが可能となる。

【0023】さらに、前記データ管理方法を提供するにあたって、更新処理の発生時に設定に基づいて更新処理種別と処理の対象となるオブジェクト、及び属性を関連付け、主データベースから従データベースへ更新データの同期を行うデータ管理手段をクライアントが有するデータ管理方法を提供する。

【0024】かかるデータ管理システムを用いることにより、データ更新時の主データベースの負荷を軽減することが可能となる。

【0025】

【発明の実施の形態】以下、本発明の実施の形態について図面を用いて詳細に説明する。以下の図中において、同一の部分には同一の符号を付与する。

【0026】本実施例においては、本発明を適用するデータベースシステムとして、主データベースとして動作する共通リポジトリにおいてデータを一元管理すると共に、共通リポジトリに対して行われた更新データを従データベースとして動作するアプリケーションサーバへ同期するデータ管理システムについて説明するが、本実施例以外にも、複数のサーバにデータを分散して管理する

分散データベースシステムに対しても適用可能である。

【0027】図1は本発明を実現するデータ管理システムの機能構成図である。

【0028】LANなどの伝送媒体22により共通リポジトリ10、アプリケーションサーバ20、クライアント40が接続されている。

【0029】共通リポジトリ10は管理するデータをDB100に記憶し、アプリケーションサーバ20は独自の処理に必要なデータをDB21に記憶する。

【0030】共通リポジトリ10は、クライアント40やアプリケーションサーバ20との通信処理を実行する通信制御部23、ユーザからの入力、及びユーザへの出力処理を制御するユーザインタフェース制御部24、同期方式に関する設定を記憶する同期管理テーブル2、受信した処理の内容を解析するアクセス要求解析部1、処理の内容に従ってDB100を更新するDB制御部25、遅延同期する更新データを記録しておく遅延更新ログ6、更新処理の解析結果、同期管理テーブル2の同期設定、及び遅延更新ログ6に記録された更新内容を基に、同期処理の整合性を保証して適切な同期方式を決定する同期整合性管理部3、アプリケーションサーバ20へ同期するデータを生成し遅延更新ログ6に対して生成、書き込み、削除などの処理を行う更新データ生成部4、生成された同期データをアプリケーションサーバ20に同期するために通信制御部23に同期データを通知する同期制御部5を有する。

【0031】アクセス要求解析部1は、共通リポジトリ10に対して要求された更新処理の発生をトリガに要求された処理の内容を解析し、更新処理であるかどうかを判別する。その結果が更新処理であった場合には、更新種別を判別する。

【0032】本実施例は、図1のような機能構造を有するデータ管理方法を実現するデータ管理システムである。

【0033】図2は本実施例のデータ管理システムのシステム構成である。

【0034】共通リポジトリ10は、中央演算装置CPU34、ハードディスクなどの2次記憶媒体（以下、磁気ディスクという）33、主記憶装置メモリ（以下、主メモリという）31、データバス32、表示装置などの出力装置（以下、ディスプレイという）35、入力装置（以下、キーボード36、マウス37という）から構成される。

【0035】図2のDB100は、データを磁気ディスク33上に格納する。また、遅延更新ログ6、同期管理テーブル2も同様に磁気ディスク33上に格納する。

【0036】主メモリ31には、通信制御プログラム23、ユーザインタフェースプログラム24、アクセス要求解析プログラム1、同期制御プログラム5、同期整合性管理プログラム3、更新データ生成プログラム4、D

B制御プログラム25が格納される。これらのプログラムは、磁気ディスク33に格納されており、必要に応じて主メモリ31に転送された後、CPU34で実行される。

【0037】続いて、本実施例における同期管理テーブル2、及び遅延更新ログ6について図3を用いて説明する。

【0038】同期管理テーブル2は、共通リポジトリ10とアプリケーションサーバ20との同期方法に関する設定を記憶するテーブルであり、1つの同期アプリケーション毎に1テーブルを有する配列構造をなし、同期対象アプリケーション数のテーブルから成る。各テーブルは同期AP管理領域200、オブジェクト更新種別管理テーブル210、同期属性管理テーブル220から構成される。

【0039】同期AP管理領域200は、同期対象アプリケーションの種別と同期処理に使用するIDを設定する記憶領域であり、同期AP種別201と識別ID202を記憶する2つの領域から構成される。同期アプリケーション種別201にはアプリケーションの種類やその名称などのデータが格納され、識別ID202には同期処理において設定を参照する時にアプリケーションの識別を可能にするために一意の整数値が格納される。

【0040】オブジェクト更新種別管理テーブル210は、更新種別とそれに対応する同期タイミングを設定するテーブルであり、前記テーブル210の各レコードは、更新種別211、及び同期方式212から構成される。更新種別211には、共通リポジトリ10のDB100に対する更新処理であるデータの追加(append)、削除(delete)の何れか1つが設定される。同期方式212は、更新種別211に対して適用する同期タイミングを設定する項目であり、即時同期または遅延同期の何れかが設定される。同期方式212には「0」か「1」が格納され、それぞれ即時同期、遅延同期を表すものとする。

【0041】同期属性管理テーブル220は、共通リポジトリ10のDB100に格納されるオブジェクトが保持する属性について、属性の種類と更新種別毎に同期タイミングを設定するテーブルである。同期属性管理テーブル220の各レコードは、同期属性種別221、更新種別222、及び同期方式223から構成される。同期属性種別221には、オブジェクトが保持する属性が設定される。例えば、名前、メールアドレス、パスワードなどである。更新種別222には、同期属性種別221に設定された属性に対する更新処理である属性の追加(append)、変更(replace)、削除(delete)が設定される。同期方式223は、同期属性種別221、及び更新種別222に設定された属性とそれに対応する更新種別に対する更新処理発生時の同期タイミングを設定する項目であり、オブジェクト更新種別管理

テーブル210の同期方式212の項目と同じである。

【0042】遅延更新ログ6は、配列構造110をなし、遅延更新処理の内容を複数格納可能である。遅延更新ログ6は、更新履歴番号の記憶領域111、更新対象エントリの名称の記憶領域112、更新種別の記憶領域113、更新内容の記憶領域114からなる。ここで、更新履歴番号111とは、各更新データを一意に識別するための整数値であり、更新処理が実行された順番に1つずつ増加していく。遅延更新ログ6内のレコードは、更新履歴番号111の小さい順に並んでいる。遅延更新ログ6のデータ例を115に示す。

【0043】次に、同期管理テーブル2の設定操作例を説明する。

【0044】同期管理テーブル2の各項目の設定操作に当たっては、ユーザは共通リポジトリ10のディスプレイ35、キーボード36、マウス37などを使用する。

【0045】ユーザは新規にアプリケーションサーバ20を同期対象とする時には、当該アプリケーションサーバ20をGUIなどで提供される設定手段を用いて登録する。この時、同期AP管理領域200内の同期AP種別201にはアプリケーション名称が登録され、新規にアプリケーション名称が登録されるに伴って、自動的に一意の整数値である識別IDが登録される。

【0046】また、ユーザはGUIなどの提供された手段を用いて同期管理テーブル2に登録されているアプリケーションサーバ20の一覧から設定を所望するアプリケーションサーバ20を選択し、オブジェクト更新種別管理テーブル210、及び同期属性管理テーブル220の内容を設定する。この設定に基づいてアプリケーションサーバ20毎に適用される同期方法が決定される。同様に、ユーザはGUIなどの手段を用いて、オブジェクト更新種別管理テーブル210については、選択したアプリケーションサーバ20に対して更新種別である追加(append)と削除(delete)に対してそれぞれ即時同期、或いは遅延同期を適用することを選択する。また、同期属性管理テーブル220については、同期対象であるアプリケーションサーバ20が管理する同期属性種別毎にその更新種別である、追加(append)、変更(replace)、削除(delete)に対してそれぞれ即時同期、遅延同期を選択する。各同期方式の項目には、即時同期として設定された項目には整数値「0」が、遅延同期に設定された項目には整数値「1」が格納される。

【0047】次に、本実施例における、共通リポジトリ10に対する更新要求発生時の処理の流れについて図12を用いて説明する。

【0048】図12は、DB100に登録されたデータの更新を要求する複数の処理が発生した時の処理シーケンスである。

【0049】クライアント40が発行したデータの追加

要求91を共通リポジトリ10が受信すると、同期整合性管理部3はその更新内容を解析して整合性判定を行う。その結果が遅延同期であった場合には更新内容を遅延更新ログ6に書き込む。また、共通リポジトリ10はDB100のデータを更新し、クライアントに追加応答92を返す。

【0050】次に、クライアント40が発行したデータの変更要求93を共通リポジトリ10が受信すると、共通リポジトリ10はDB100のデータを更新すると共に、同期整合性管理部3がその更新内容を解析して整合性判定を行う。その結果が即時同期であった場合には更新内容と遅延更新ログ6を読み込み、更新データ生成部4がこれらの要求を一括して同期するための同期データを生成する。この時、即時同期処理である変更要求93を同期するためには遅延同期処理として遅延更新ログ6に記録されている追加要求91が必要な処理であるという関連性が検出され、追加要求91は変更要求93の即時同期に伴い一括して同期処理が行われる。

【0051】従って、共通リポジトリ10からアプリケーションサーバ20へ追加要求94と変更要求95が要求され、アプリケーションサーバ20は受信した更新要求に従ってデータの更新を行い、共通リポジトリ10へ追加応答96と変更応答97を返す。共通リポジトリ10は、アプリケーションサーバ20からの前記応答96、97を受けてクライアント40へ変更応答98を返す。

【0052】この時、共通リポジトリ10のログにおいては、追加要求91、変更要求93のそれぞれについてクライアント40より更新要求を受信してDB100を更新した時点で、その記録がログに書き込まれていることが確認できる。一方、アプリケーションサーバ20においては、共通リポジトリ10からの同期要求として追加要求94、及び変更要求95が一括して受信されるため、アプリケーションサーバ20のログを参照することによって、2つの更新要求94、95が一括して処理されたことが確認可能である。

【0053】従って、これらログを参照することによって、共通リポジトリ10に対する要求発生においては時間差が存在した2つの更新要求91、93に関して、変更要求93の発生をトリガに同要求93に伴って追加要求91に関しても同期処理がなされ、アプリケーションサーバ20においては追加要求94及び変更要求95が一括して同期データとして反映され、DB21に対するデータ更新が行われることは明らかである。

【0054】また、本実施例の提供する機能については、製品などにおいては一般にマニュアルなどに記載されることからも本発明の適用は明らかである。

【0055】以下に、本実施例におけるデータ管理システムの動作を説明する。

【0056】図4は、クライアント40から共通リポジ

トリ10へ発行された要求の解析に関する動作を表すフローチャートである。

【0057】共通リポジトリ10の通信制御部23がクライアント40からの要求を受信すると、アクセス要求解析部1が更新要求であるか否かを判別し（S401）、受信した要求が更新要求であれば、要求された更新種別により処理を分岐する（S402からS404）。

【0058】更新要求種別の解析の結果、追加要求であれば追加処理（S405）、変更要求であれば変更処理（S406）、削除要求であれば削除処理（S407）、前記以外の処理である場合は各自に対応した処理を実行する（S408）。

【0059】図5は、追加処理（図4のS405）の動作を表すフローチャートである。

【0060】まず、同期整合性管理部3はステップS501において同期管理テーブル2を取得し、ステップS502において、DB制御部25は更新要求に従って共通リポジトリ10のDB100へオブジェクトを追加する。

【0061】次に、同期整合性管理部3は、取得した同期管理テーブル2内の先頭のテーブルに記載の同期アプリケーションに対して、ステップS503においては要求された更新種別に対応する同期設定に基づいて処理を分岐する。更新種別が即時同期処理の対象か、遅延同期処理の対象であるかについては、ステップS401で取得した更新種別を基にしてステップS501で取得した同期管理テーブル2内のオブジェクト更新種別管理テーブル210を参照することによって判定する。

【0062】即時同期を適用する設定の場合、ステップS505において同期整合処理を実行して整合性を保証した同期データを生成する。そして、ステップS506においては生成された同期データを同期制御部5がアプリケーションサーバ20へ同期する。

【0063】一方、ステップS503において遅延同期設定として判定された場合、ステップS504において更新データ生成部4は遅延更新ログに当該更新情報を追加する。

【0064】上記一連の処理終了後、ステップS507において同期管理テーブル2を基に未だ同期処理を行っていない同期対象アプリケーションが存在するか否かを判定し、存在する場合にはステップS503から同様の処理を繰り返し実行する。

【0065】図6は、変更処理（図4のS406）の動作を表すフローチャートである。

【0066】まず、ステップS601において同期整合性管理部3は同期管理テーブル2を取得し、ステップS602において共通リポジトリ10のDB制御部25は更新要求に従ってDB100に格納された対象オブジェクトデータを変更する。

【0067】次に、同期整合性管理部3は、取得した同期管理テーブル2の先頭のテーブルに記載された同期アプリケーションに対して、ステップS603においては要求された処理内容と遅延更新ログの内容を基に遅延更新最適化処理を行う。

【0068】続いて、ステップS604においては、要求された更新種別による同期設定に基づいて処理を分岐する。更新種別が即時同期処理の対象か、或いは遅延同期処理の対象であるかについては、ステップS401で取得した更新種別を基にしてステップS601で取得した同期管理テーブル2のオブジェクト更新種別管理テーブル210を参照することによって判定する。

【0069】即時同期を適用する設定の場合、ステップS606において同期整合処理を実行して整合性を保証した同期データを生成する。そして、ステップS607においては、ステップS606において生成された同期データを同期制御部5がアプリケーションサーバ20へ同期する。

【0070】一方、ステップS604において遅延同期設定と判定された場合、ステップS605において更新データ生成部4は遅延更新ログ6に当該更新情報を追加する。

【0071】上記一連の処理終了後、ステップS608において同期管理テーブル2を基に未だ同期処理を行っていない同期対象アプリケーションが存在するか否かを判定し、存在すればステップS603から同様の処理を実行する。

【0072】図7は、削除処理(図4のS407)の動作を表すフローチャートである。

【0073】まず、ステップS701において同期整合性管理部3は同期管理テーブル2を取得し、ステップS702においては共通リポジトリ10のDB制御部25が更新要求に従ってDB100に格納された対象オブジェクトデータを削除する。次に、同期整合性管理部3はステップS703において、取得した同期管理テーブル2の先頭のテーブルに記載された同期アプリケーションに対して、要求された処理内容と遅延更新ログ6の内容を基に遅延更新最適化処理を行う。

【0074】続いて、ステップS704においては、要求された更新種別に対応した同期設定に基づいて処理を分岐する。更新種別が即時同期処理の対象か、或いは遅延同期処理の対象であるかについては、ステップS401で取得した更新種別を基にしてステップS701で取得した同期管理テーブル2のオブジェクト更新種別管理テーブル210を参照することによって判定する。

【0075】即時同期を適用する設定の場合、ステップS706において処理内容を基に更新データ生成部4が生成した同期データを同期制御部5がアプリケーションサーバ20へ同期する。

【0076】一方、ステップS704において遅延同期

設定と判定された場合、ステップS705において更新データ生成部4は遅延更新ログに当該更新情報を追加する。

【0077】上記一連の処理終了後、ステップS707において同期管理テーブル2を基に未だ同期処理を行っていない同期対象アプリケーションが存在するか否かを判定し、存在すればステップS703から同様の処理を実行する。

【0078】図8は、同期整合処理(図5のS505、図6のS606)の動作を表すフローチャートである。

【0079】同期整合性管理部3は、ステップS801では遅延更新ログ6を読み込み、ステップS802においてステップS401で取得した更新対象のオブジェクトと同一のオブジェクトに対する遅延更新データが遅延更新ログ6内に記録されているか否かを検索する。

【0080】次に、ステップS803においてはステップS802で実行した検索処理の結果を用いて、処理対象として同一オブジェクトが存在するか否かを判定する。

【0081】同一オブジェクトが存在する場合、ステップS804において遅延更新ログ6から該当するログを抽出し、続いてステップS805において前記遅延更新ログ6より抽出した更新種別が追加処理であるか変更処理であるかを判定する。

【0082】ステップS805における判定の結果が追加処理であった場合、ステップS806において関連オブジェクト操作処理を行う。

【0083】一方、ステップS805における判定の結果が変更処理であった場合、ステップS807においては、ステップS401とステップS804において取得した変更内容を基に同一属性に対する変更処理が存在するか否かを判定する。

【0084】ステップS807において変更処理の対象に同一属性が存在すると判定された場合、ステップS808において遅延更新データ内の該当データを削除する。一方、同一属性が存在しないと判定された場合は特別に処理は行わない。

【0085】また、ステップS803において同一オブジェクトが存在しないと判定された場合、ステップS809において関連オブジェクト操作処理を行う。

【0086】そして、最後にステップS810において更新データ生成部4が同期データを生成する。

【0087】図9は、関連オブジェクト操作処理(図8のS806、S809、図9のS903)の動作を表すフローチャートである。

【0088】まず、同期整合性管理部3は、ステップS901において、ステップS401で取得した更新処理の内容とステップS801で取得した遅延更新ログ6の内容から更新対象のオブジェクト間の関連性を取得する。ここで言う関連性について、本実施例のデータ管理

システムの一例としてディレクトリシステムを挙げて説明する。

【0089】ディレクトリシステムにおいては人、組織、ネットワーク機器などサービスに関する情報を木構造で管理するため、或るエントリを追加するためにはその親となるエントリが存在する必要があるということが挙げられる。ディレクトリツリー構造を現実の組織階層構造に模して設計する場合、或る企業組織において総務部庶務課に所属するユーザ「山田」をエントリとして追加するためには、組織「庶務課」がエントリとして既にディレクトリ上に存在していなければならない。これをDNで表現すると、ユーザ「山田」のDNは一例として「cn=山田, ou=庶務課, ou=総務部...」として表現され、組織「庶務課」のDNは「ou=庶務課, ou=総務部...」として表現される。従って、DNにおける「ou=庶務課, ou=総務部...」の部分に関しては両エントリにおいて共通の名称となる。ステップS901においては、このような関係を検出して更新内容の関連性を取得することを行う。

【0090】次に、ステップS901の結果を基に、ステップS902においては関連オブジェクトが存在するかどうかを判定する。

【0091】関連オブジェクトが存在すると判定された場合、ステップS901において検出された関連エントリに対して、ステップS903において関連オブジェクト操作を実行する。

【0092】ステップS904においては、更新データ生成部4は遅延更新ログ6に記録された当該関連オブジェクトに関する更新内容を基に同期データを生成する。そして、ステップS905においては、ステップS904において生成された同期データの内容を遅延更新ログ6より削除する。

【0093】一方、ステップS902において関連オブジェクトが存在しないと判定された場合にはそのまま関連オブジェクト操作処理を終了する。

【0094】図10は、遅延更新最適化処理（図6のS603、図7のS703）の動作を表すフローチャートである。

【0095】まず、ステップS1001においては、遅延更新ログ6を読み込み、ステップS1002においてはステップS401で取得した更新処理の対象オブジェクトと同一のオブジェクトに対する更新処理が遅延更新ログ6内に存在するかを検索する。

【0096】次に、ステップS1003においては、ステップS1002の結果を基に同一オブジェクトに対する更新処理が存在するか否かを判定する。

【0097】判定の結果、同一オブジェクトに対する更新が遅延更新ログ6に記録されている場合、ステップS1004においてはステップS401で判定した更新種別を基に処理を分岐する。

【0098】ステップS1004において変更処理と判定された場合には、ステップS1005において遅延更新ログ6より更新処理の内容を抽出する。

【0099】さらに、ステップS1006においては、ステップS1004において抽出した変更内容と同一の属性に対する変更が記録されているか否かを判定する。

【0100】判定の結果同一の属性が存在する場合、ステップS1007において該当する更新データを遅延更新ログ6から削除する。一方、ステップS1006において変更処理の対象に同一の属性がないと判定された場合には、そのまま遅延更新最適化処理を終了する。

【0101】また、ステップS1003において同一オブジェクトに対する更新処理が存在しないと判定された場合、及びステップS1004において更新種別が削除処理であると判定された場合についても、同様に遅延更新最適化処理を終了する。

【0102】次に、遅延同期処理について説明する。遅延同期処理は共通リポジトリ10内に蓄積された遅延更新ログ6の内容を、あらかじめ設定された遅延同期条件に基づいてアプリケーションサーバ20に同期する処理である。

【0103】図11は、遅延同期処理の動作を表すフローチャートである。

【0104】遅延同期処理ではあらかじめ設定された遅延同期条件が満たされた時に、共通リポジトリ10からアプリケーションサーバ20へ同期処理が行われる。遅延同期条件とは、例えば以下のようないくつかの条件が設定可能である。（1）時間条件。時間、日、週、月、年単位などの決まった時間単位毎に同期を実行する条件であり、例えば毎週日曜日の22:00から同期処理を開始するといった条件設定が挙げられる。（2）データ量。遅延更新ログ6の更新データのレコード数が一定数以上になった時、遅延更新ログのファイルサイズが一定サイズ以上の大きさになった時などの条件が挙げられる。設定可能な同期条件は拡張可能であり、前記2つの条件以外の条件も適用可能である。

【0105】同期制御部5は、ステップS1101において同期条件が満たされたか否かを監視し、同期設定条件が満たされたと判定した時、ステップS1102においては更新データ生成部4が遅延更新ログ6を取得して同期データを生成する。

【0106】ステップS1102において生成された同期データについては、ステップS1103において同期制御部5がアプリケーションサーバ20へ同期する。なお、同期対象となる遅延同期情報は、遅延更新ログ6に記録された全ての更新データである。

【0107】最後に、ステップS1103の同期処理の終了後、ステップS1104において、更新データ生成部4は遅延更新ログ6を共通リポジトリ10上から削除する。遅延更新ログ6は、遅延同期処理が実行される度

に削除され、再び共通リポジトリ10に更新処理が発生することで生成される。

【0108】以上で本発明のデータ管理方法における第1の実施例を説明した。かかるデータ管理方法によって、主データベースから従データベースへのデータ同期を行う際に、更新要求の種別、また更新処理の対象となる属性についての同期タイミングの関連付けを可能にすると共に、オブジェクト間の関連付けを可能にすることによって、主データベースと従データベース間のデータ整合性を保証する同期機能を提供することが可能である。

【0109】次に、第2の実施例この実施例は理解できない。アプリケーションサーバは検索要求を受けたときに何故対象の同期が遅延されていることがわかるのか？について、図13から図15を用いて説明する。但し、第1の実施例との相違点に限り説明する。

【0110】第1の実施例においては、即時同期を適用する更新内容と遅延同期としてログに記録された更新データを関連付けて、遅延データも併せて即時に同期するための同期データの生成、同期を行うことを可能としたが、本実施例においては、即時同期を適用する更新要求について遅延更新ログ6のデータと関連付けて併せて遅延同期の対象として、同期条件が満たされた時に一括して同期する機能を提供するデータ管理システムである。

【0111】図15は、共通リポジトリ10のDB100に登録されたデータの更新を要求する複数の処理が発生した時の処理シーケンスである。

【0112】クライアント40が発行したデータの追加要求81を共通リポジトリ10が受信すると、同期整合性管理部3はその更新内容を解析して整合性判定を行う。その結果が遅延同期であった場合には更新内容を遅延更新ログ6に書き込む。また、共通リポジトリ10のDB100のデータを更新し、クライアントに追加応答82を返す。

【0113】次に、クライアント40が発行したデータの変更要求83を共通リポジトリ10が受信すると、共通リポジトリ10のDB100のデータを更新すると共に、同期整合性管理部3はその更新内容を解析して整合性判定を行う。その結果が即時同期であった場合には更新内容と遅延更新ログ6を読み込み、変更要求83に関連する更新処理が既に遅延更新ログ6に記録されていた時には、アプリケーションサーバ20へは即時に同期せず、変更要求83の更新内容を遅延更新ログ6に書き込むと共に、クライアント40に変更応答84を返す。

【0114】共通リポジトリ10において同期条件が満たされると、更新データ生成部4は遅延更新ログ6を基に同期データを生成し、同期制御部5はアプリケーションサーバ20へ遅延更新データを同期するために追加要求85と変更要求86を発行する。

【0115】アプリケーションサーバ20は共通リポジ

トリ10からの追加要求85及び変更要求86を受信すると、要求に従って、DB21への追加処理と変更処理を行い、共通リポジトリ10へ追加応答87と変更応答88を返す。

【0116】本実施例における各種処理に関わる動作の流れは第1の実施例とほぼ同様である。

【0117】但し、同期整合処理において同期整合処理、及び関連オブジェクト操作処理が第1の実施例と異なる。

【0118】図13は、本実施例における同期整合処理の動作を表すフローチャートである。

【0119】ステップS1303において、要求された更新処理の対象であるオブジェクトが遅延更新ログ6に記録された遅延更新に存在するか否かを判定する処理までは第1の実施例と同様である。

【0120】次に、ステップS1303において同一オブジェクトが存在すると判定された場合には、同期整合性管理部3によって即時同期として判定された即時にアプリケーションサーバ20へ同期する処理を行わず、更新要求を遅延更新ログ6に書き込む。

【0121】一方、ステップS1303において同一オブジェクトが存在しないと判定された場合には、ステップS1304において関連オブジェクト操作処理を実行する。

【0122】図14は、本実施例における関連オブジェクト操作処理の動作を表すフローチャートである。

【0123】図14のステップS1403において関連オブジェクト操作処理が行われるところまでは、第1の実施例と同様である。

【0124】ステップS1404においては、ステップS1402において関連オブジェクトが存在すると判定されているため、同期整合性管理部3によって即時同期として判定された更新要求を遅延更新ログ6に書き込む。

【0125】以上説明した第2の実施例によると、即時に同期する設定がなされた更新要求に対して、共通リポジトリ10において遅延同期を行うために遅延更新ログ6に記録され更新データと関連付け共通リポジトリ10からアプリケーションサーバ20への同期条件が満たされた時に、併せて遅延同期処理を行うことで、データベース間のデータ整合性を保証する同期機能を提供することが可能となる。

【0126】次に、第3の実施例について説明する。但し、第1の実施例との相違点に限り説明する。

【0127】本実施例は、第1の実施例において共通リポジトリ10に組み込まれた更新データの整合性を管理する機能をクライアント40に組み込んだデータ管理システムである。本実施例のデータ管理システムは、第1の実施例と異なり、クライアントがデータベースシステムに対する全てのデータ更新を行う。

【0128】本実施例のデータ管理システムの機能構成は、第1の実施例で説明した図1において、クライアント40が通信制御部23、ユーザインタフェース制御部24、アクセス要求解析部1を有する。また、図3で説明した同期管理テーブル2、遅延更新ログ6、同期整合性管理部3、更新データ生成部4、同期制御部5については、共通リポジトリ10は備えず、クライアント40が有する。

【0129】本実施例のクライアント40のシステム構成は、第1の実施例における図2における構成とほぼ同様である。

【0130】但し、HDD33にDB100が存在しない点、及び主メモリ31にDB制御プログラム25が存在しない点において異なる。本実施例の共通リポジトリ10のシステム構成は、主メモリ31にアクセス要求解析プログラム1、DB制御プログラム25、通信制御プログラム23のみが格納される点、及び磁気ディスク33にDBのみが格納される点を除けば、図2で示した第1の実施例と同様である。

【0131】次に、第2の実施例における、データ管理システム全体の処理の流れについて、クライアント40において共通リポジトリ10のデータの更新を要求する複数の処理が発生した時の処理シーケンスを説明する。

【0132】ユーザがクライアント40を通じて共通リポジトリ10への追加要求を入力し、次に変更処理を要求した時に、同期データを生成するまでは第1の実施例と同様の処理をクライアント40において行う。

【0133】同期データ生成後、クライアント40はアプリケーションサーバ20へ同期要求を出し、アプリケーションサーバ20は受信した同期要求に従ってデータの更新を行い、クライアント40へ同期応答を返す。

【0134】以下に、本実施例におけるデータ管理システムの動作を説明する。

【0135】本実施例における、クライアント40において要求された処理の解析に関する動作は第1の実施例の図4と同様である。

【0136】前記解析の結果、追加処理が要求された場合、図5とほぼ同様の処理を行うが、ステップS502のオブジェクトの追加処理にあたっては、共通リポジトリ10への追加要求を発行する。

【0137】変更処理が要求された場合、及び削除処理が要求された場合についても同様の処理を行う。

【0138】以上説明した第3の実施例によると、共通リポジトリ10とアプリケーションサーバ20の同期データの生成にわたる整合性判定処理、及び同期データ生成処理を実行する機能をクライアント40に持たせることで、データ更新時の共通リポジトリ10の負荷を軽減することが可能となる。

### 【0139】

【発明の効果】以上述べたように本発明によれば、複数データベース間のデータ管理において、管理するデータベース毎に更新対象属性と更新種別を基にそれぞれ同期タイミングを設定し、データの更新処理発生時にシステムが更新内容であるオブジェクト、及び属性間の関連付けを行うことによって同期タイミングの相違に関わらず整合性を保証したデータ管理が可能である。そのため、ユーザがデータベース間で新旧の異なるデータを使用するために誤って業務を遂行してしまう状況に陥ることが無くなり、また、不整合発生に伴うシステム管理者の整合性回復作業を不要とすることでき、データベースシステムの利便性の向上を実現可能である。

#### 【図面の簡単な説明】

【図1】本発明の第1の実施例の機能構成図である。

【図2】本発明の第1の実施例のシステム構成図である。

【図3】本発明の同期管理テーブル、及び遅延更新ログのデータ構造図である。

【図4】本発明の実施例のアクセス要求解析処理にわたる動作の説明図である。

【図5】本発明の実施例の追加処理にわたる動作の説明図である。

【図6】本発明の実施例の変更処理にわたる動作の説明図である。

【図7】本発明の実施例の削除処理にわたる動作の説明図である。

【図8】本発明の実施例の同期整合処理にわたる動作の説明図である。

【図9】本発明の実施例の関連オブジェクト操作処理にわたる動作の説明図である。

【図10】本発明の実施例の遅延更新最適化処理にわたる動作の説明図である。

【図11】本発明の実施例の遅延同期処理にわたる動作の説明図である。

【図12】本発明の第1の実施例の同期整合処理を伴うデータ処理における処理シーケンスを示す図である。

【図13】本発明の第2の実施例の同期整合処理にわたる動作の説明図である。

【図14】本発明の第2の実施例の関連オブジェクト操作処理にわたる動作の説明図である。

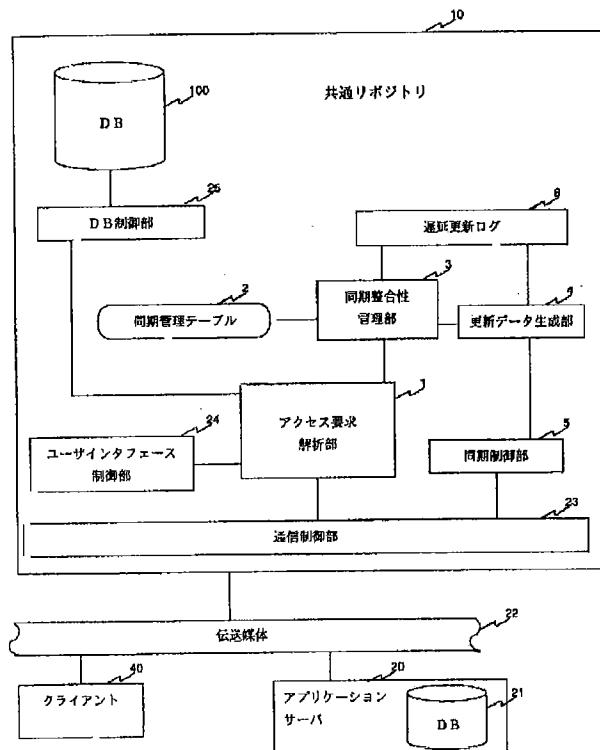
【図15】本発明の第2の実施例の同期整合処理を伴うデータ処理における処理シーケンスを示す図である。

#### 【符号の説明】

1…アクセス要求解析部、2…同期管理テーブル、3…同期整合性管理部、4…更新データ生成部、5…同期制御部、6…遅延更新ログ、10…共通リポジトリ、20…アプリケーションサーバ。

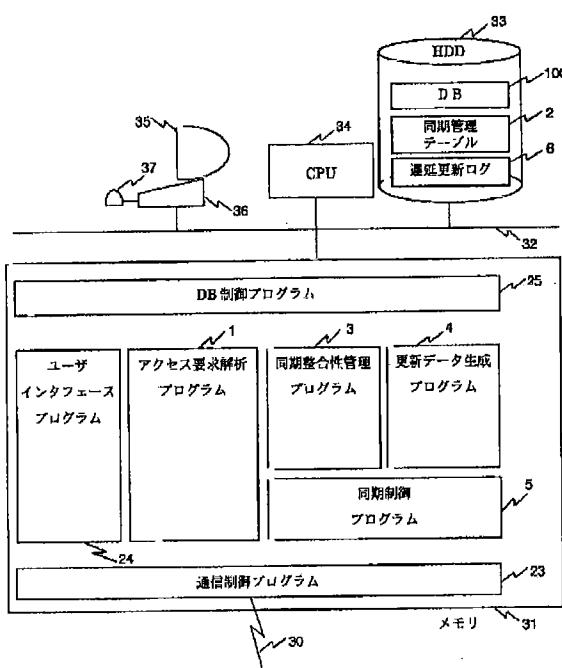
【図1】

図 1



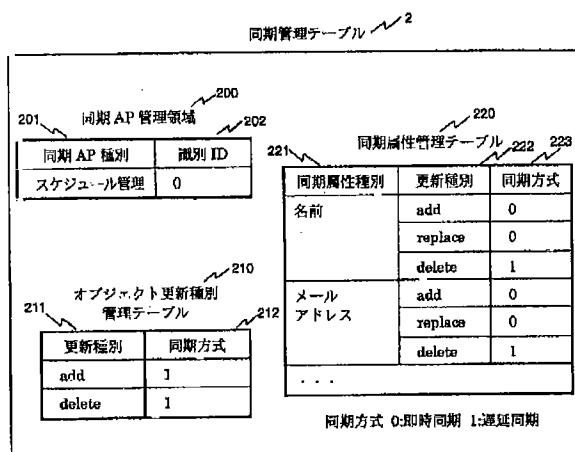
【図2】

図 2



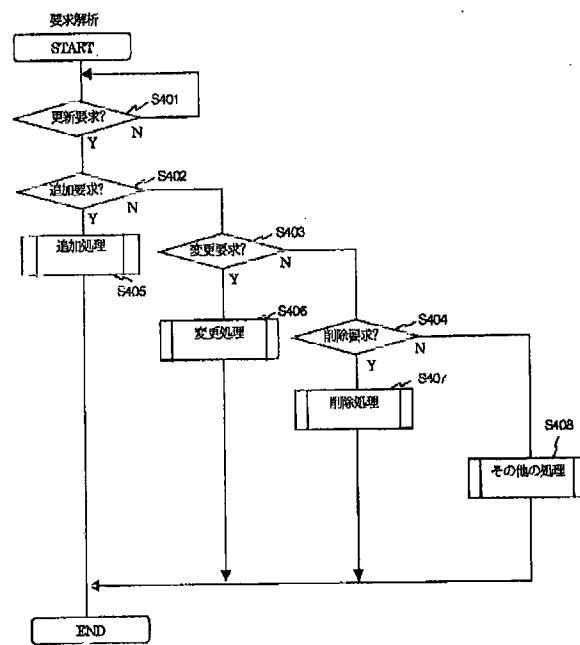
【図3】

図 3



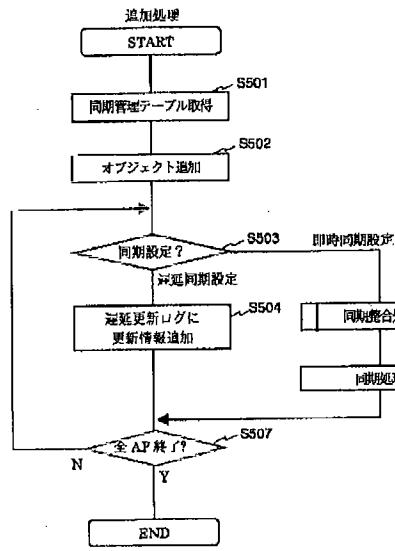
【図4】

図 4



【図5】

図 5

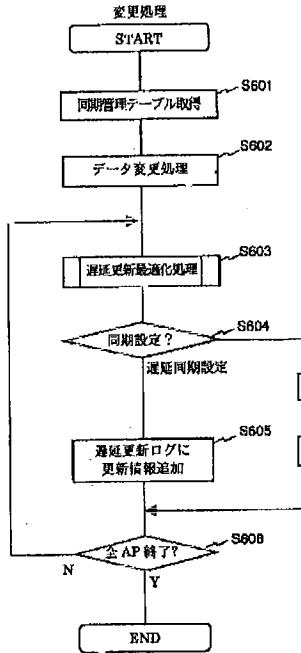


【図7】

図 7

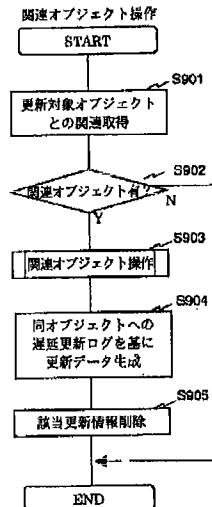
【図6】

図 6



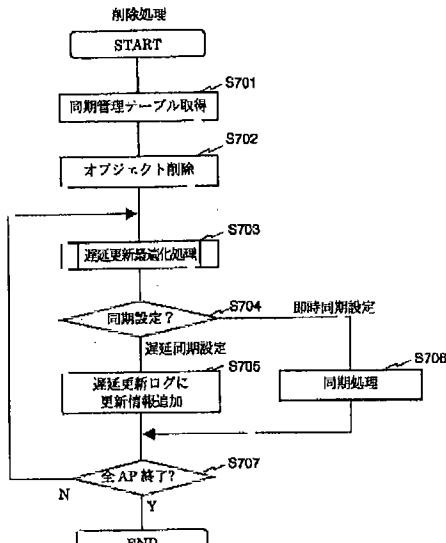
【図9】

図 9



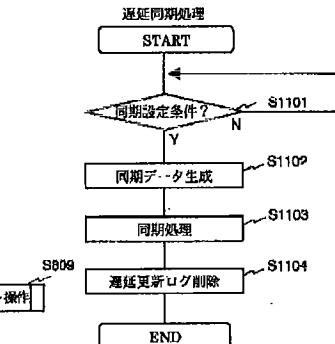
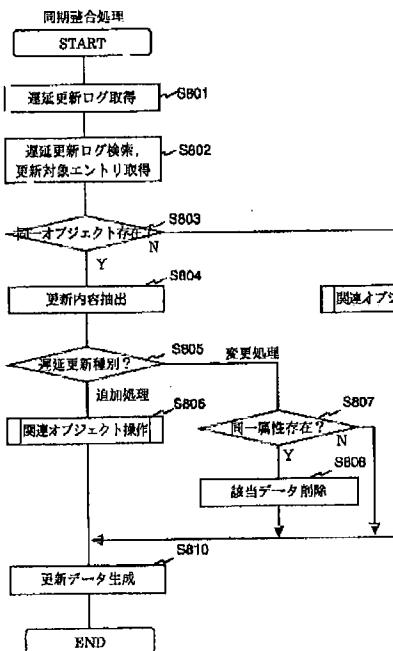
【図8】

図 8



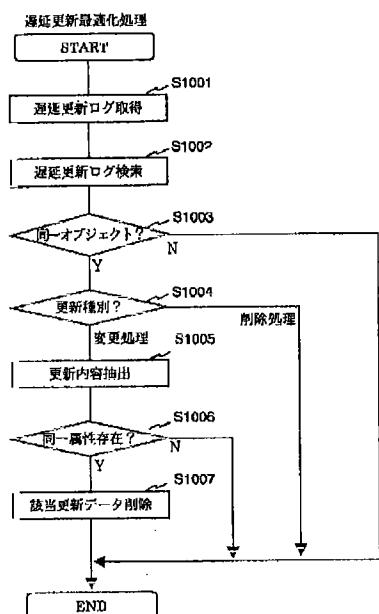
【図11】

図 11



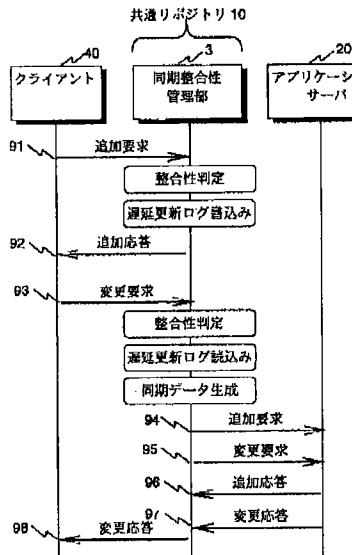
【図10】

図 10



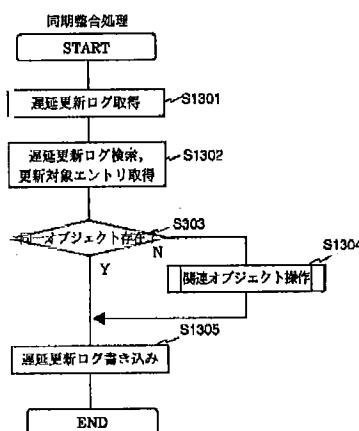
【図12】

図 12



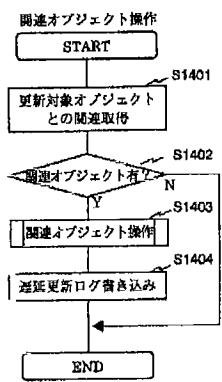
【図13】

図 13



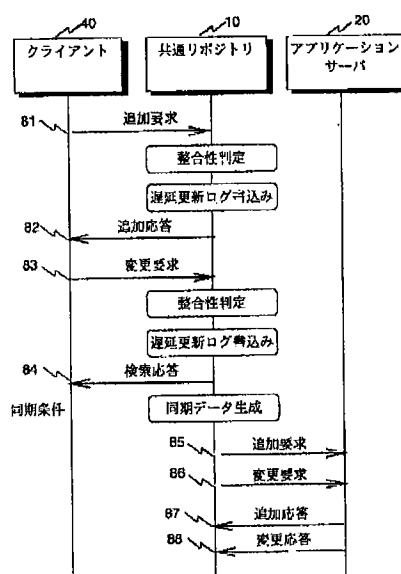
【図14】

図 14



【図15】

図 15



フロントページの続き

(72)発明者 志賀 賢太

神奈川県川崎市麻生区王禅寺1099番地 株  
式会社日立製作所システム開発研究所内

(72)発明者 由井 仁

神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

F ターム(参考) 5B075 KK04 NR02 PP02 PP03 PQ02  
PQ05 PR03  
5B082 AA01 GA14 HA03